Terms used
private and **global** and **registers** and **VLIW** and **floating point** and **execution unit**

Found **52,596** of **131,734**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200

Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Compiler transformations for high-performance computing
David F. Bacon, Susan L. Graham, Oliver J. Sharp
December 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 4

Full text available: pdf(6.32 MB)

Additional Information: full citation, abstract, references, citings, index terms, review

In the last three decades a large number of compiler transformations for optimizing programs have been implemented. Most optimizations for uniprocessors reduce the number of instructions executed by the program using transformations based on the analysis of scalar quantities and data-flow techniques. In contrast, optimizations for high-performance superscalar, vector, and parallel processors maximize parallelism and memory locality with transformations that rely on tracking the properties o ...

Keywords: compilation, dependence analysis, locality, multiprocessors, optimization, parallelism, superscalar processors, vectorization

2 Computing curricula 2001
September 2001 **Journal on Educational Resources in Computing (JERIC)**

Full text available: pdf(613.63 KB) html(2.78 KB)

Additional Information: full citation, references, citings, index terms

3 Software pipelining loops with conditional branches
Mark G. Stoodley, Corinna G. Lee
December 1996 **Proceedings of the 29th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.64 MB) Publisher Site

Additional Information: full citation, abstract, references, citings, index terms

Software pipelining is an aggressive scheduling technique that generates efficient code for loops and is particularly effective for VLIW architectures. Few software pipelining algorithms, however, are able to efficiently schedule loops that contain conditional branches. We have developed an algorithm we call All Paths Pipelining (APP) that addresses this shortcoming of software pipelining. APP is designed to achieve optimal or near-optimal performance for any run of iterations while providing ef ...

4 Evaluation of the WM architecture
Wm. A. Wulf

h c g e cf c

April 1992 **ACM SIGARCH Computer Architecture News , Proceedings of the 19th annual international symposium on Computer architecture**, Volume 20 Issue 2

Full text available: pdf(897.06 KB) Additional Information: full citation, abstract, references, citings, index terms

This report describes the results of studies of the WM architecture—its performance, the values of some of its key architectural parameters, the difficulty of compiling for it, and hardware implementation complexity. The studies confirm that, with comparable chip area and without heroic compiler technology, WM is capable of outperforming traditional scalar architectures by factors of 2-9. They also underscore the need to devise higher bandwidth memory systems.

5 A survey of processors with explicit multithreading

Theo Ungerer, Borut Robič, Jurij Šilc

March 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 1

Full text available: pdf(920.16 KB) Additional Information: full citation, abstract, references, index terms

Hardware multithreading is becoming a generally applied technique in the next generation of microprocessors. Several multithreaded processors are announced by industry or already into production in the areas of high-performance microprocessors, media, and network processors.A multithreaded processor is able to pursue two or more threads of control in parallel within the processor pipeline. The contexts of two or more threads of control are often stored in separate on-chip register sets. Unused i ...

Keywords: Blocked multithreading, interleaved multithreading, simultaneous multithreading

6 Initial results on the performance and cost of vector microprocessors

Corinna G. Lee, Derek J. DeVries

December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.73 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Increasingly wider superscalar processors are experiencing diminishing performance returns while requiring larger portions of die area dedicated to control rather than datapath. As an alternative to using these processors to exploit parallelism effectively, we are investigating the viability of using single-chip vector microprocessors. This paper presents some initial results of our investigation where we compare the performance and cost of vector microprocessors to that of aggressive, out-of-or ...

7 MOVE: a framework for high-performance processor design

Henk Corporaal, Hans (J.M.) Mulder

August 1991 **Proceedings of the 1991 ACM/IEEE conference on Supercomputing**

Full text available: pdf(1.04 MB) Additional Information: full citation, references, citings, index terms

8 An elementary processor architecture with simultaneous instruction issuing from multiple threads

Hiroaki Hirata, Kozo Kimura, Satoshi Nagamine, Yoshiyuki Mochizuki, Akio Nishimura, Yoshimori Nakase, Teiji Nishizawa

April 1992 **ACM SIGARCH Computer Architecture News , Proceedings of the 19th annual international symposium on Computer architecture**, Volume 20 Issue 2

Full text available: pdf(1.03 MB) Additional Information: full citation, abstract, references, citings, index terms

In this paper, we propose a multithreaded processor architecture which improves machine throughput. In our processor architecture, instructions from different threads (not a single

h c g e cf c

thread) are issued simultaneously to multiple functional units, and these instructions can begin execution unless there are functional unit conflicts. This parallel execution scheme greatly improves the utilization of the functional unit. Simulation results show that by executing two and four threads in parallel ...

9 Exploiting horizontal and vertical concurrency via the HPSm microprocessor

Wen-Mei W. Hwu, Yale N. Patt

December 1987 **Proceedings of the 20th annual workshop on Microprogramming**

Full text available: pdf(801.30 KB) Additional Information: full citation, abstract, references, citings

HPSm is a single-chip microarchitecture designed and implemented at the University of California to achieve high performance. The approach is to exploit both vertical and horizontal concurrency in the microarchitecture. Experiments have been conducted to demonstrate the effectiveness of HPSm as compared to a popular single-chip microarchitecture, the Berkeley RISC/SPUR. Evaluations have been done with both control intensive and floating point intensive benchmarks. For both types of benchmar ...

10 Architectural and organizational tradeoffs in the design of the MultiTitan CPU

N. P. Jouppi

April 1989 **ACM SIGARCH Computer Architecture News , Proceedings of the 16th annual international symposium on Computer architecture**, Volume 17 Issue 3

Full text available: pdf(1.32 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes the architectural and organizational tradeoffs made during the design of the MultiTitan, and provides data supporting the decisions made. These decisions covered the entire space of processor design, from the instruction set and virtual memory architecture through the pipeline and organization of the machine. In particular, some of the tradeoffs involved the use of an on-chip instruction cache with off-chip TLB and floating-point unit, the use of direct-mapped instead o ...

11 Multiscalar processors

Gurindar S. Sohi, Scott E. Breach, T. N. Vijaykumar

May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture**, Volume 23 Issue 2

Full text available: pdf(1.44 MB) Additional Information: full citation, abstract, references, citings, index terms

Multiscalar processors use a new, aggressive implementation paradigm for extracting large quantities of instruction level parallelism from ordinary high level language programs. A single program is divided into a collection of tasks by a combination of software and hardware. The tasks are distributed to a number of parallel processing units which reside within a processor complex. Each of these units fetches and executes instructions belonging to its assigned task. The appearance of a single log ...

12 Software support for speculative loads

Anne Rogers, Kai Li

September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international conference on Architectural support for programming languages and operating systems**, Volume 27 Issue 9

Full text available: pdf(1.33 MB) Additional Information: full citation, references, citings, index terms

13 KCM: a knowledge crunching machine

H. Benker, J. M. Beacco, M. Dorochevsky, Th. Jeffré, A. Pöhlmann, J. Noyé, B. Poterie, J. C. Syre, O. Thibault, G. Watzlawik

April 1989 **ACM SIGARCH Computer Architecture News , Proceedings of the 16th annual international symposium on Computer architecture**, Volume 17 Issue 3

Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, citings, index

h c g e cf c

terms

KCM (Knowledge Crunching Machine) is a high-performance back-end processor which, coupled to a UNIX* desk-top workstation, provides a powerful and user-friendly Prolog environment catering for both development and execution of significant Prolog applications. This paper gives a general overview of the architecture of KCM stressing some new features like a 64-bit tagged architecture, shallow backtracking and an original memory management unit. Some early benchmark result ...

14 Multiscalar processors
Gurindar S. Sohi, Scott E. Breach, T. N. Vijaykumar
August 1998 **25 years of the international symposia on Computer architecture (selected papers)**
Full text available: pdf(1.57 MB) Additional Information: full citation, references, citings, index terms

15 Exploiting idle floating-point resources for integer execution
S. Subramanya Sastry, Subbarao Palacharla, James E. Smith
May 1998 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1998 conference on Programming language design and implementation**, Volume 33 Issue 5
Full text available: pdf(1.65 MB) Additional Information: full citation, abstract, references, citings, index terms

In conventional superscalar microarchitectures with partitioned integer and floating-point resources, all floating-point resources are idle during execution of integer programs. Palacharla and Smith [26] addressed this drawback and proposed that the floating-point subsystem be augmented to support integer operations. The hardware changes required are expected to be fairly minimal.To exploit these idle floating resources, the compiler must identify integer code that can be profitably offloaded to ...

16 Simultaneous multithreading: maximizing on-chip parallelism
Dean M. Tullsen, Susan J. Eggers, Henry M. Levy
May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture**, Volume 23 Issue 2
Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper examines *simultaneous multithreading,* a technique permitting several independent threads to issue instructions to a superscalar's multiple functional units in a single cycle. We present several models of simultaneous multithreading and compare them with alternative organizations: a wide superscalar, a fine-grain multithreaded processor, and single-chip, multiple-issue multiprocessing architectures. Our results show that both (single-threaded) superscalar and fine-grain multithr ...

17 Cycles to recycle: garbage collection to the IA-64
Richard L. Hudson, J. Elliot Moss, Sreenivas Subramoney, Weldon Washburn
October 2000 **ACM SIGPLAN Notices , Proceedings of the second international symposium on Memory management**, Volume 36 Issue 1
Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, citings, index terms

The IA-64, Intel's 64-bit instruction set architecture, exhibits a number of interesting architectural features. Here we consider those features as they relate to supporting garbage collection (GC). We aim to assist GC and compiler implementors by describing how one may exploit features of the IA-64. Along the way, we record some previously unpublished object scanning techniques, and offer novel ones for object allocation (suggesting some simple operating system support that would simplify it ...

18 Improving single-process performance with multithreaded processors
Alexandre Farcy, Olivier Temam
January 1996 **Proceedings of the 10th international conference on Supercomputing**

h c g e cf c

Full text available: pdf(1.07 MB) Additional Information: full citation, references, index terms

19 Converting thread-level parallelism to instruction-level parallelism via simultaneous multithreading

Jack L. Lo, Joel S. Emer, Henry M. Levy, Rebecca L. Stamm, Dean M. Tullsen, S. J. Eggers
August 1997 **ACM Transactions on Computer Systems (TOCS)**, Volume 15 Issue 3

Full text available: pdf(526.39 KB) Additional Information: full citation, abstract, references, citings, index terms, review

To achieve high performance, contemporary computer systems rely on two forms of parallelism: instruction-level parallelism (ILP) and thread-level parallelism (TLP). Wide-issue super-scalar processors exploit ILP by executing multiple instructions from a single program in a single cycle. Multiprocessors (MP) exploit TLP by executing different threads in parallel on different processors. Unfortunately, both parallel processing styles statically partition processor resources, thus preventing t ...

Keywords: cache interference, instruction-level parallelism, multiprocessors, multithreading, simultaneous multithreading, thread-level parallelism

20 Superscalar architectures: Reducing the complexity of the register file in dynamic superscalar processors

Rajeev Balasubramonian, Sandhya Dwarkadas, David H. Albonesi
December 2001 **Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.34 MB) Additional Information: full citation, abstract, references, citings
Publisher Site

Dynamic superscalar processors execute multiple instructions out-of-order by looking for independent operations within a large window. The number of physical registers within the processor has a direct impact on the size of this window as most in-flight instructions require a new physical register at dispatch. A large multi-ported register file helps improve the instruction-level parallelism (ILP), but may have a detrimental effect on clock speed, especially in future wire-limited technologies. ...

h c g e cf c